Testing numeric functions

Here's an idiom to check whether a function call `f(ARGUMENT.LIST)` returns the expected `answer`:

```
stopifnot(isTRUE(all.equal(answer, f(ARGUMENT.LIST))))
```

e.g. `stopifnot(isTRUE(all.equal(1.414, sqrt(2))))`

Background:

- Using `==` to test equality between real numbers (`numeric` in R) is unreliable. e.g.

  ```
  49*(1/49) == 1
  ```

- `all.equal(x, y, tolerance=(.Machine$double.eps ^ 0.5))`, returns `TRUE` if `x` is close to `y` (that is, if `abs(x - y) < tolerance`). Otherwise it returns an error message. e.g.

  ```
  all.equal(49*(1/49), 1)
  all.equal(0, 1) # it would be convenient if this were FALSE
  ```

- `isTRUE(x)` returns `TRUE` if `x` is `TRUE`, and `FALSE` otherwise.

- `stopifnot(...)` stops if any logical expression in `...` is not all TRUE. e.g.

  ```
  stopifnot(0 < 1)
  stopifnot(0 < 1, 1:3 < 3)
  ```